



Attorney Docket No. YOR920000470US1

# 5  
605  
3-15-04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): A. Leff et al.  
Docket No.: YOR920000470US1  
Serial No.: 09/703,187  
Filing Date: October 31, 2000  
Group: 2155  
Examiner: Young N. Won

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Signature: *Ben C. Leff*

Date: March 2, 2004

Title: System and Method for Dynamic Runtime  
Partitioning of Model-View-Controller Applications

TRANSMITTAL OF FORMAL DRAWINGS

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Attention: Official Draftsperson

Sir:

RECEIVED

MAR 11 2004

Technology Center 2100

Applicants submit herewith seven (7) sheets of formal drawings in the above-referenced patent application.

Respectfully submitted,

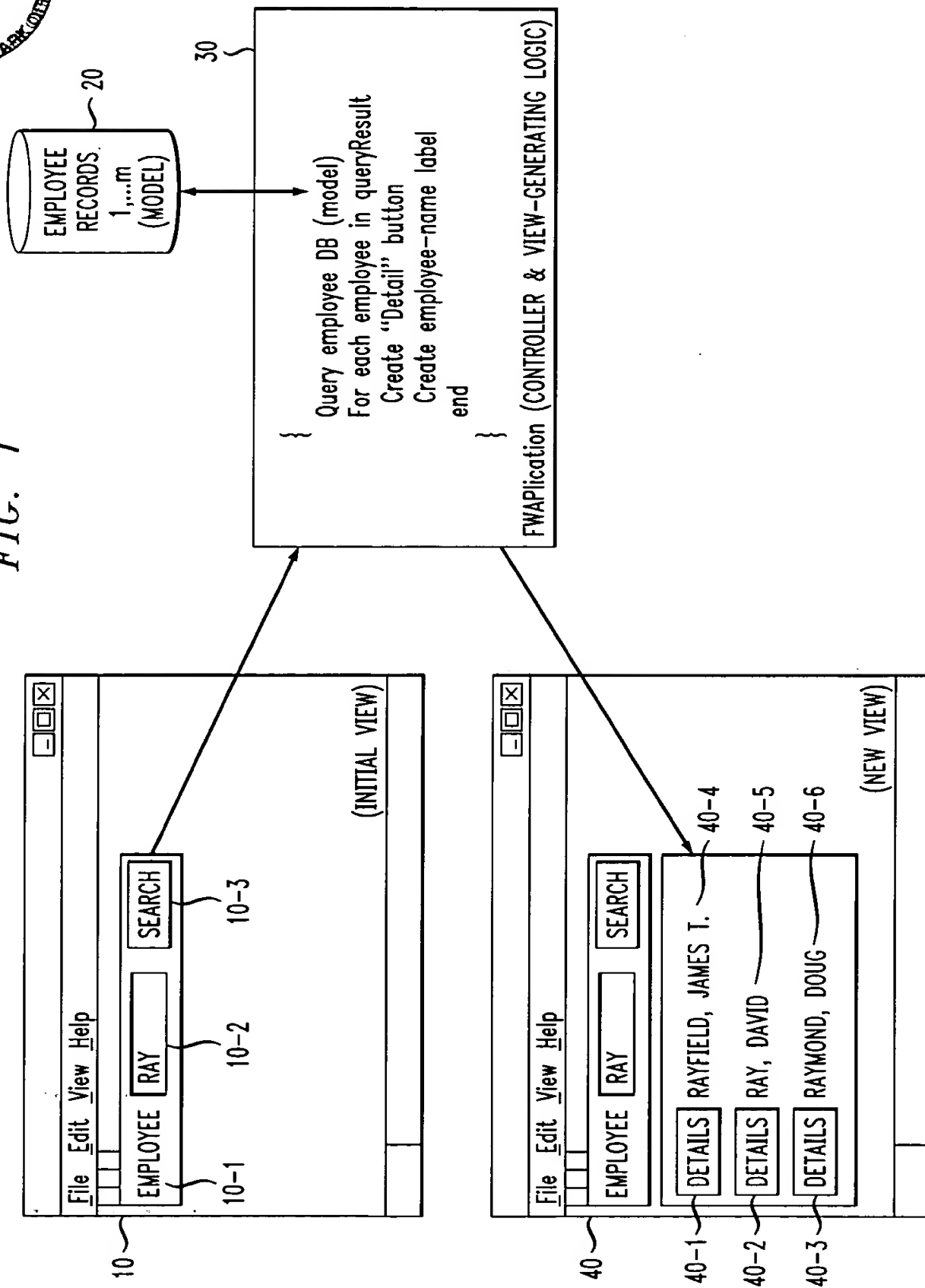
*William E. Lewis*

William E. Lewis  
Attorney for Applicant(s)  
Reg. No. 39,274  
Ryan, Mason & Lewis, LLP  
90 Forest Avenue  
Locust Valley, NY 11560  
(516) 759-2946

Date: March 2, 2004



FIG. 1





*FIG. 2*

11  
}

```
public interface ViewAPI
{
11-1 → public ViewElement createElement(ViewType viewType, String name);
11-2 → public void remove ViewElement(String name);
11-3 → public ViewElement getViewElement(String name);
}

public interface ViewElement
{
11-4 → public ViewProperty getViewProperty(String name);
11-5 → public void setViewProperty(String name, PropertyValue value);
11-6 → public void registerController
        (Controller controller, ControllerArguments args, UserInteraction event);
}
```

*FIG. 3*

12  
}

```
public interface ModelAPI
{
12-1 → public ModelHome locateHome (String homeName);
}

public interface ModelHome
{
12-2 → public ModelElement createModelElement (ModelElementKey key);
12-3 → public void removeModelElement (ModelElementKey key);
12-4 → public ModelElementSet Query (QueryCriteria criteria);
}

public interface ModelElement
{
12-5 → public ModelElementKey getKey();
}
```



FIG. 4

30

```
// Controller

30-0 → public void ProcessSearchRequest
    {
    // access the view components
30-10 → TextField  inputName = ViewAPI.getViewElement(inputKey);
30-20 → String    employeeName = inputName.getProperty("data");

    // access the model
30-30 → EmployeeHome employeeHome = ModelAPI.locateHome("employees");
    // query the model
30-40 → Employee[] results =
        employeeHome.Query (new QueryCriteria ("findByName," employeeName));

    // generate the next view, and return to client
30-50 → setView(BuildSearchResultsView (name, results));
    }

    // View Generation code

30-60 → public void CreateInitialView ()
    {
    // create the top-level frame
30-70 → Frame    frame = ViewAPI.createViewElement ("frame," frameKey);
30-80 → TextField inputName = ViewAPI.createViewElement ("textfield," inputKey);

    // associate a controller with user-input into the textfield
30-90 → inputName.registerController (ProcessSearchRequest, null, TextField.CHANGE);
    }
```

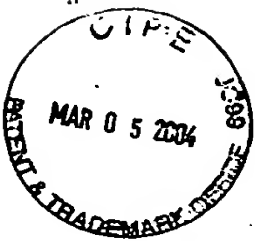


FIG. 5

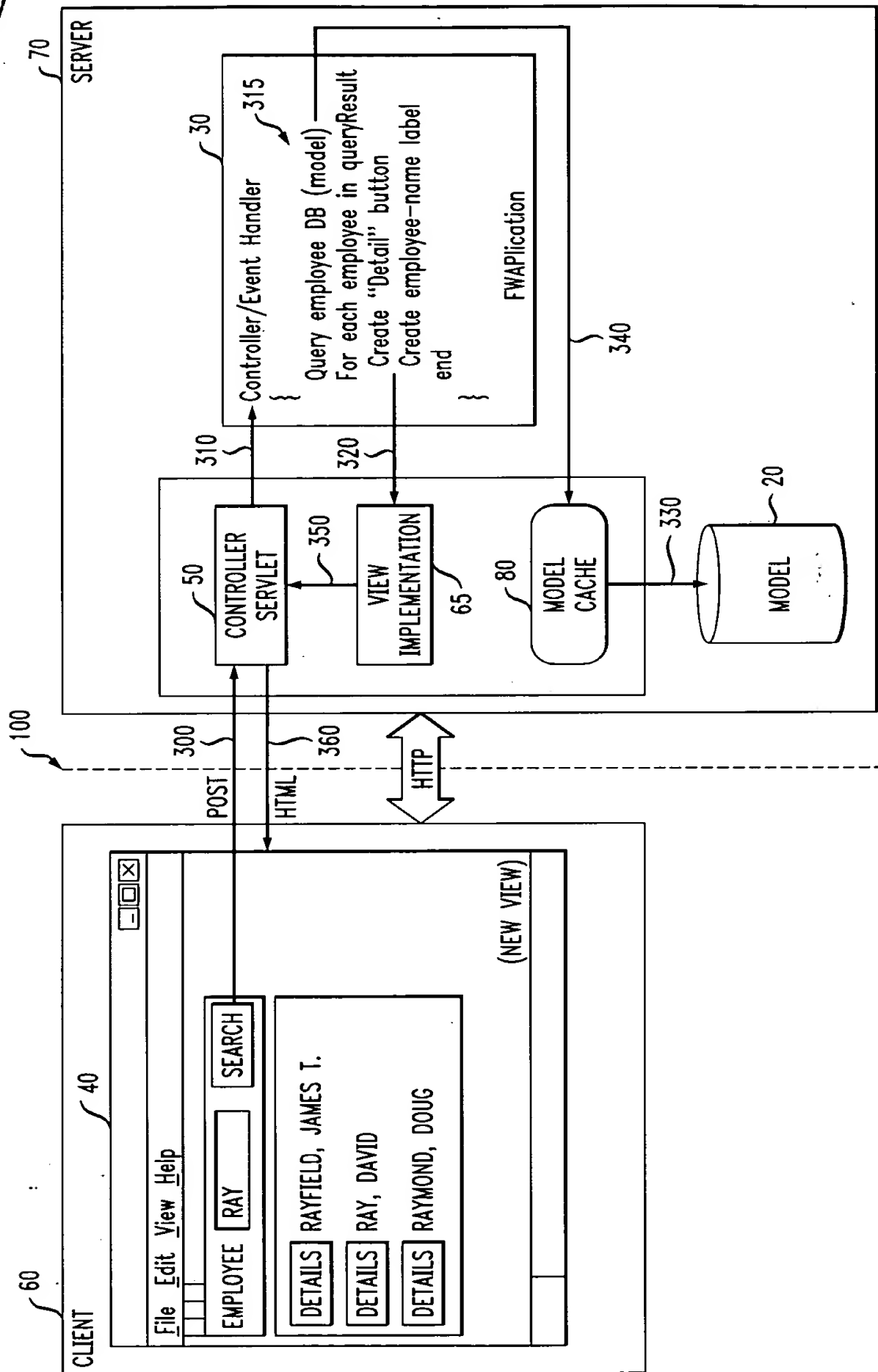


FIG. 6

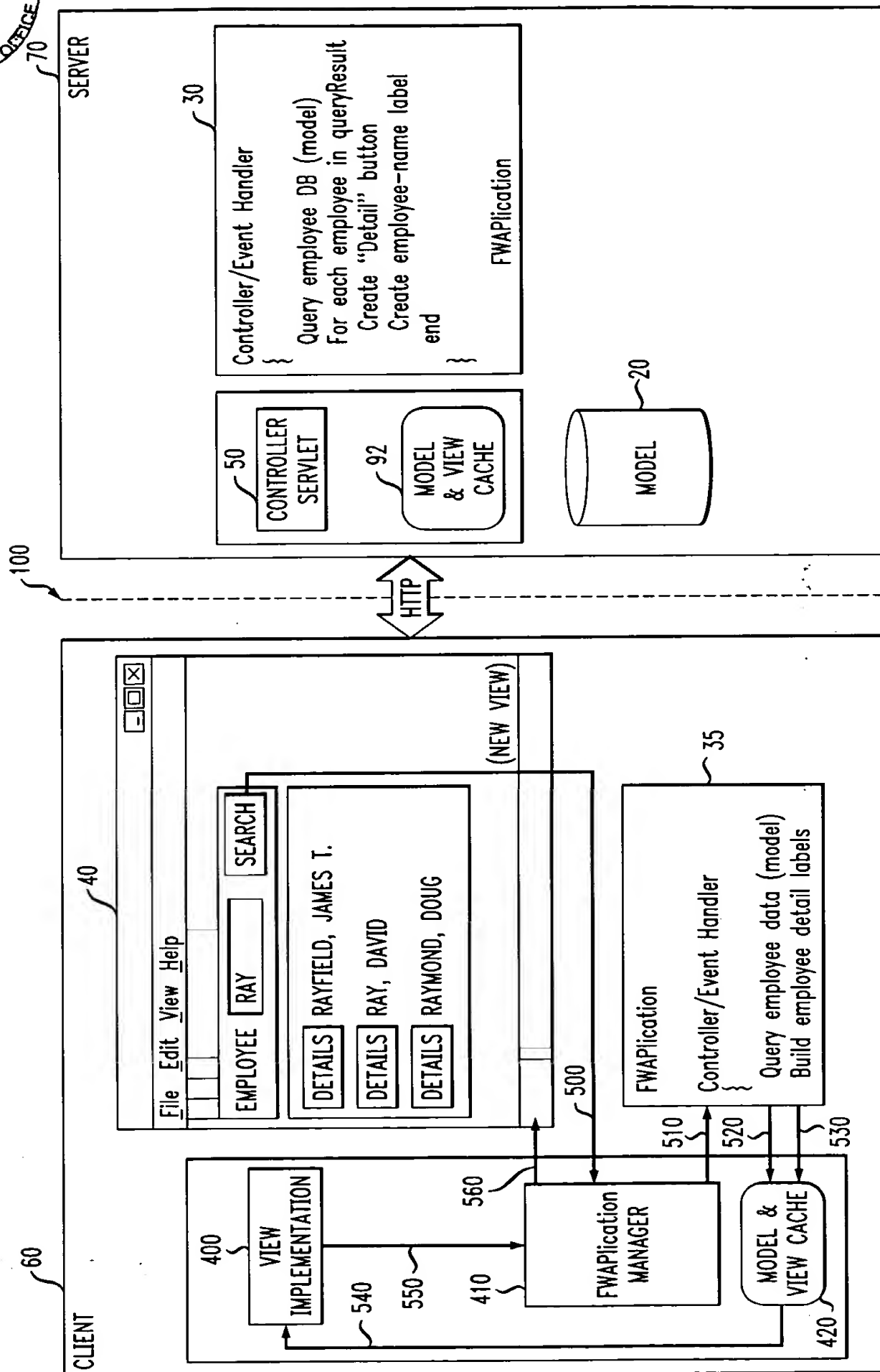
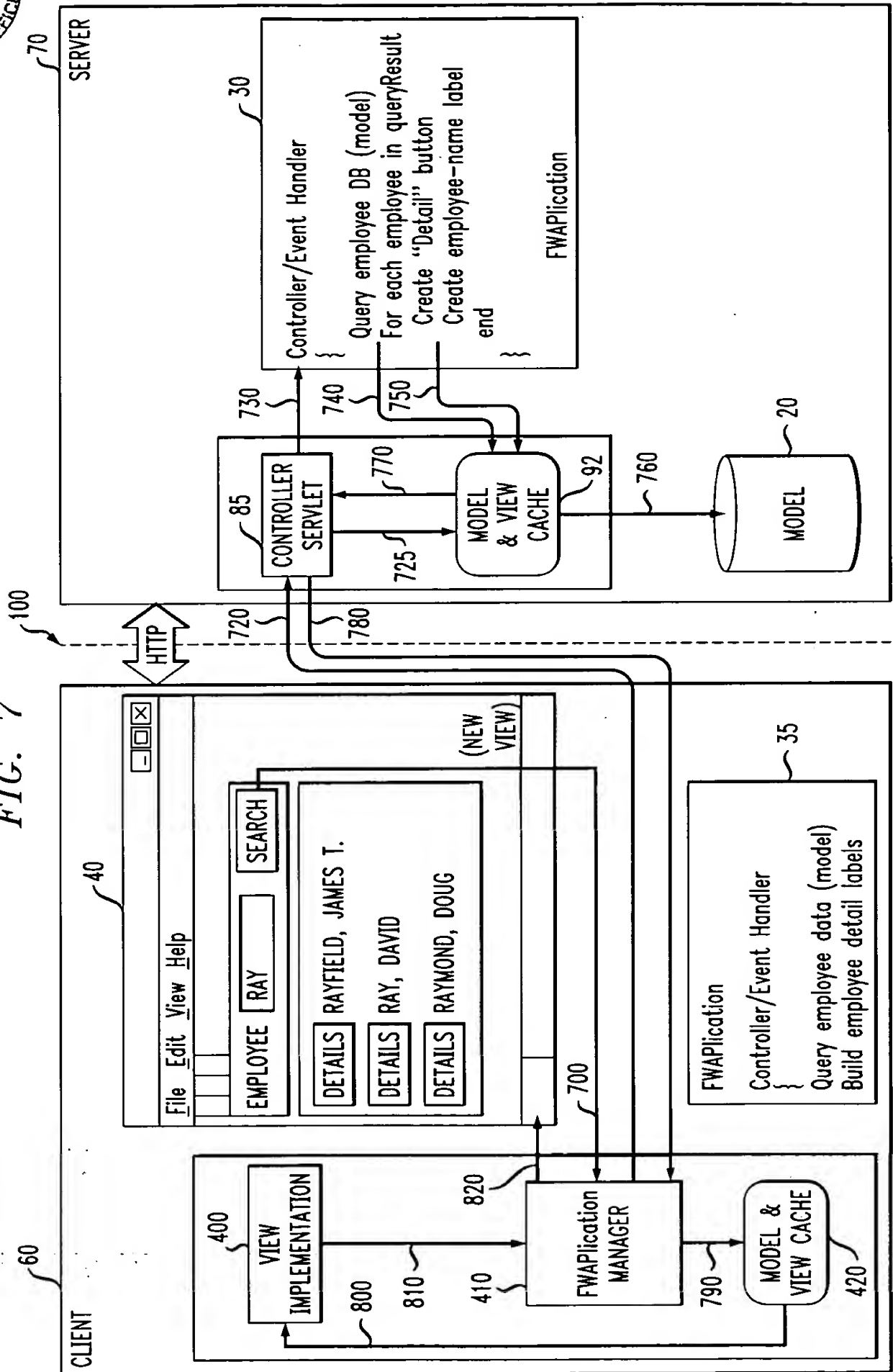




FIG. 7





*FIG. 8*

